Analisis *High Availability* Pada Sistem Berbasis Teknologi *Oracle Data Guard* (Studi Kasus SIA-SAT UKSW)

Kristoko Dwi Hartomo, T. Arie Setiawan P., Sandy Pratama
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia
Email: kristoko@staff.uksw.edu

Abstract. High Availability on Oracle Data Guard-based System (Case Study: SIA-SAT UKSW). *SIA-SAT is a subject administration system for UKSW collegian. Database SIA-SAT is backuped by autobackup system, and backup file sent directly to backup server at another building. Until these days, database SIA-SAT system still does not have a secondary database (backup) which can replace main database when failure is happening on main database.Oracle Data Guard System is a technology researched by Oracle to increase availability level on Oracle Database server. Every transaction is saved on main server and duplicated for standby database server. The advantages point of technology is standby database which indirectly becomes a backup server, and standby database can be used for replacing main database.The result of this research is the technology of Oracle Data Guard can increase availability level for a system Oracle Database. This condition can happen because standby database always duplicates data and is ready to replace primary database.*

Keywords: *availability, data guard, database distribution*

Abstrak. SIA-SAT adalah sebuah sistem administrasi mata kuliah untuk mahasiswa. Basis data SIA-SAT dicadangkan oleh sistem cadang otomatis, dan berkas cadangan tersebut dikirimkan secara langsung ke server cadangan yang terletak di gedung lain. Hingga saat ini, sistem basis data SIA-SAT masih belum memiliki basis data sekunder (cadangan) yang dapat menggantikan basis data utama jika kegagalan terjadi pada basis data utama. *Oracle Data Guard System* adalah sebuah teknologi yang diteliti oleh *Oracle* untuk meningkatkan tingkat ketersediaan pada *Oracle Database server*. Setiap transaksi disimpan pada server utama dan diduplikasi untuk *standby database server*. Manfaat dari teknologi ini adalah *standby database* secara tidak langsung menjadi server cadangan, dan *standby database* dapat digunakan untuk menggantikan basis data utama. Hasil dari penelitian ini adalah teknologi Oracle Data Guard dapat meningkatkan tingkat ketersediaan untuk sistem *Oracle Database*. Kondisi ini dapat tercapai karena *standby database* selalu menduplikasi data dan siap untuk menggantikan basis data primer.

Kata Kunci: ketersediaan, *data guard*, distribusi basis data

1. Pendahuluan

Basis data terkomputerisasi merupakan hal yang penting dalam organisasi modern. Masing-masing dari kita sebenarnya selalu berhubungan dengan basis data dalam kegiatan sehari-hari seperti: berbelanja di toko, transaksi keuangan pada *Automated Teller Machine* (ATM), memesan barang secara *online*, dan registrasi mata kuliah. Sebagian besar hidup kita adalah bagian dari perkembangan basis data terkomputerisasi dan teknologi basis data.

Teknologi basis data tidak hanya meningkatkan kinerja organisasi namun juga kualitas pengambilan keputusan yang mempengaruhi hidup orang lain. Basis data menyimpan sebuah aliran data tentang banyak aspek seperti: data diri seseorang, pemakaian pulsa telpon, histori kredit, dan masih banyak lagi. Karena itu, basis data dapat disebut hal yang sangat penting dikarenakan menyimpan banyak data-data penting yang tidak boleh hilang.

Ada banyak hal yang dapat menyebabkan data hilang seperti: *bad sector* pada *hard disk*, listrik yang kurang stabil, program yang rusak (*corrupt*), virus komputer, kesalahan *user* waktu menghapus data, dan lain-lain. Dengan situasi seperti, teknologi *backup* dan *recovery* muncul untuk mengatasi hal-hal tersebut. Kerusakan fisik maupun non-fisik pada server basis data dapat dikembalikan ke kondisi semula dengan teknologi *backup*.

UKSW adalah salah satu perguruan tinggi yang telah menerapkan teknologi basis data sebagai pendukung kinerja operasional sehari-hari, dari urusan administrasi sampai registrasi mata kuliah untuk mahasiswa. Sehingga, data tersebut akan bernilai sangat tinggi, dan sangat penting. Untuk menjaga data tersebut, maka akan dilakukan proses *backup*. Namun, ada satu hal yang penting lagi, yaitu faktor ketersediaan *(availability)*. Jika sebuah server basis data dapat dikembalikan kapan saja dari *backup* yang ada, tetapi akan memakan waktu yang cukup lama. Untuk itulah dibutuhkan sebuah sistem yang dapat menjaga ketersediaan layanan basis data untuk sistem SIA-SAT.

Dhanu Indira Bharwara adalah salah satu yang pernah meneliti tentang *high availability*, judul penelitiannya adalah "*Cluster High Availability* Pada Server LTSP" yang bertujuan untuk menangani komputasi server dengan *thin client*. Baik server utama dan cadangan, keduanya memiliki aplikasi yang memonitoring keadaan masing-masing server, jika ada komputer yang rusak, aplikasi monitor akan langsung mengaktifkan server cadangan.

Teknologi *high availability* adalah teknologi yang dapat diterapkan pada server apa saja, dalam penelitian Bharwara, *high availability* di implementasikan pada *web server*, agar mempunyai fungsionalitas yang tinggi. *High availability* digunakan untuk menggantikan *web server* yang satu saat server tersebut rusak.

High availability berbeda dengan *load balancing*, meskipun jumlah komputer yang digunakan sama untuk menambah kemampuan, namun server pada *load balancing* bekerja bersama-sama, sedangkan pada *high availability*, setiap server akan bekerja satu-satu, *standby server* akan bekerja pada saat server primer *down*.

2. Tinjauan Pustaka

2.1. Basis Data

Basis data adalah sebuah kumpulan data yang *persistent* yang dapat dibagi dan saling berhubungan menurut Mannino. Data yang *persistent* dapat diartikan sebagai data yang disimpan dalam tempat yang aman atau stabil. Aman dalam arti data yang disimpan tidak dapat berubah atau hilang kecuali data tersebut memang diubah atau dihapus. Stabil mempunyai arti data disimpan dalam media yang tidak mudah rusak. Dapat dibagi berarti sebuah basis data mempunyai kemampuan untuk melayani banyak pengguna dan mempunyai banyak kegunaan (Saputra). Saling berhubungan adalah data dapat disimpan dalam unit yang berbeda, namun dapat terhubung menjadi sebuah gambaran data yang sempurna.

Basis data semakin berkembang, sampai saat ini basis data harus dapat menyimpan data yang berukuran *gigabyte*, dikarenakan kebutuhan yang semakin tinggi. Disamping itu, semakin tinggi kebutuhan penyimpanan akan menyebabkan tingkat penggunaan meningkat yang menyebabkan persentase kerusakan akan menjadi lebih tinggi. Pada saat terjadi kerusakan yang menyebabkan basis data *down*, saat itulah tingkat *high availability* dari suatu server menurun. Hal ini disebabkan banyak pengguna yang harus menunggu hingga basis data tersebut dapat dipergunakan kembali.

2.2. *High Availability*

High availability adalah sebuah desain protokol sistem dan implementasi dari sebuah jaminan tentang sebuah kepastian dari kelanjutan operasional dalam sebuah periode yang ditentukan (Koopmann). Kata *availability* mengacu kepada kemampuan sekelompok pengguna agar dapat mengakses ke dalam sistem seperti menambah pekerjaan baru, mengubah pekerjaan

yang sebelumnya, atau mengumpulkan hasil dari pekerjaan yang sebelumnya. Jika pengguna tidak dapat mengakses sistem, dapat dikatakan sistem tersebut tidak tersedia *(unavailable)*. Secara umum variabel yang digunakan untuk memutuskan sebuah sistem *available* atau *unavailable* adalah *downtime*.

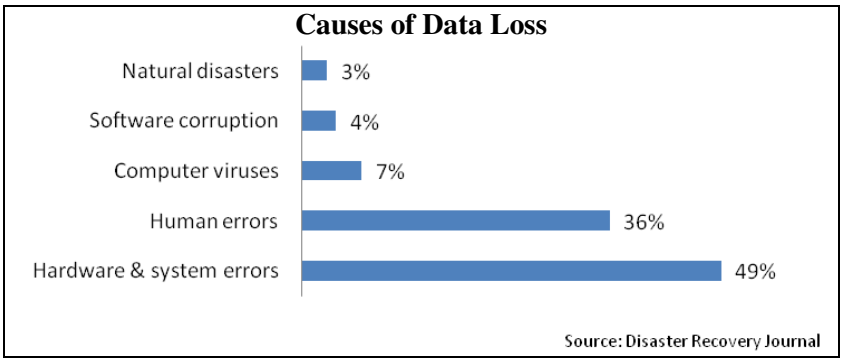
Rumus 1. Avaliability

$$availability = \frac{uptime(MTBF)}{uptime(MTBF) + downtime(MTTR)}$$

Tabel 1. Pengukuran *Availability* (Han)

Uptime **(persen)**	***Downtime*** **/ Tahun**
97.9494	7.3 hari
98.9747	3.65 hari
99.4874	43.8 jam
99.7949	17.52 jam
99.8975	8.76 jam
99.9898	52.3 menit

Pada rumus 1, *availability* adalah satuan waktu sistem untuk memulihkan diri jika terjadi *failure*. MTBF (*Mean Time Between Failure*) adalah waktu rata-rata antara *failure* yang ke *n* dengan *failure* ke *n-1*. MTTR (*Mean Time To Recover / Repair*) adalah waktu rata-rata sistem yang dibutuhkan untuk pulih dari keadaan *failure.* Tabel 1 adalah tabel pengukuran *availability*.

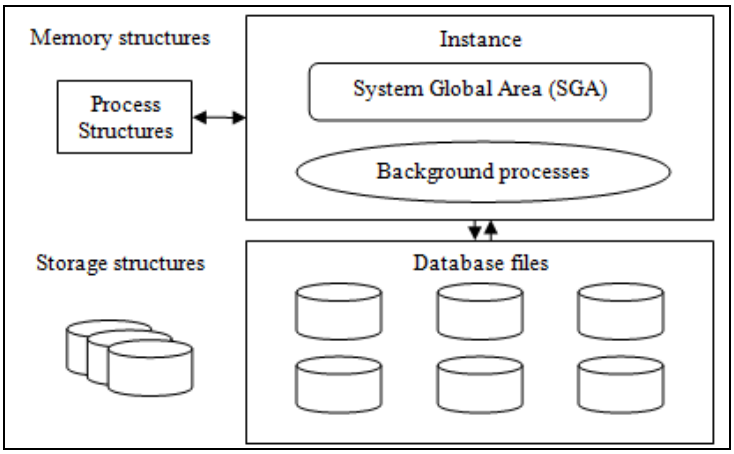


Gambar 1. Persentase Penyebab Kehilangan Data

2.3. *Oracle Database*

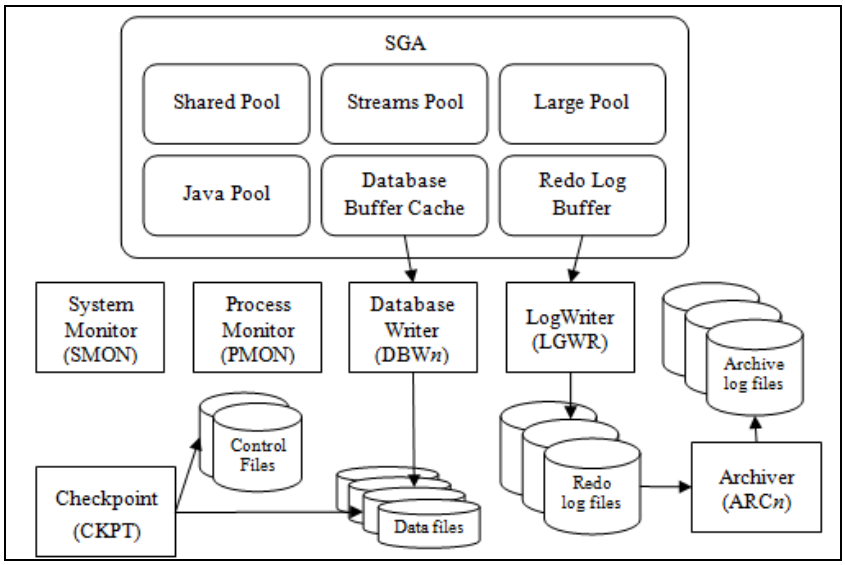
Oracle terbagi menjadi 2 bagian utama, yaitu *memory structure* dan *storage structure*. Saat sebuah basis data dijalankan dalam sebuah server basis data, *Oracle* akan mengalokasikan sejumlah *shared memory* yang disebut *System Global Area* (SGA) dan menjalankan sejumlah *background process*. Gabungan antara SGA dan *background process* disebut *Oracle instance*.

Arsitektur *Oracle* terbagi menjadi 2 bagian besar yaitu *instance* dan *database file* (Greenberg). *Instance* merupakan kumpulan proses-proses yang bekerja dibelakang *Oracle*. *Instance* bekerja untuk mengatur pemakaian prosesor, *memory*, dan *monitoring Oracle*. *Database file* bekerja sebagai tempat penyimpanan *Oracle*. Penyimpanan data, informasi, pengaturan, akan disimpan dengan baik, dan diatur oleh *instance*. *Instance* terbagi menjadi 2 bagian. Yang pertama adalah SGA. SGA befungsi sebagai kumpulan *memory* yang menjadi hal yang utama bagi *instance*. Bagian yang kedua adalah *background process* yang berfungsi untuk menjalankan *Oracle*.



Gambar 2. Arsitektur *Oracle Database*

Struktur *memory Oracle* atau pada bagian sebelumya disebutkan sebagai SGA, terbagi menjadi 6 bagian utama yang mempunyai tugas penting seperti berikut (Best): (1) *Database buffer cache*. Berisi *data block* dari *query* yang diberikan oleh *user*, setiap *user* melakukan *query SELECT* maupun *INSERT*, *UPDATE*, *DELETE*, *data block* yang berhubungan dengan *query* tersebut akan dipindahkan ke *buffer cache*. *Buffer cache* akan dituliskan ke dalam *data file* saat LGWR bekerja. (2) *Redo log buffer*. Menyimpan informasi *redo* sampai ditulis ke dalam *file physical redo log* yang disimpan didalam *disk*. *Redo* yang disimpan adalah *redo* dari perintah DML yang diberikan *user*. *Redo log* akan dikosongkan oleh LGWR saat terjadi *checkpoint*. (3) *Shared pool*. Menyimpan berbagi macam *construct* yang dapat di *share* diantara *user*. (4) *Large pool*. Bagian yang fakultatif yang menyediakan alokasi memori yang besar untuk proses-proses yang besar, seperti operasi *backup* dan *recovery*, dan proses I/O. (5) *Java pool*. Digunakan untuk semua *session* terutama kode *Java* dan data dalam *Java Virtual Machine* (JVM). (6) *Stream pool*. Digunakan untuk *Oracle Streams*.

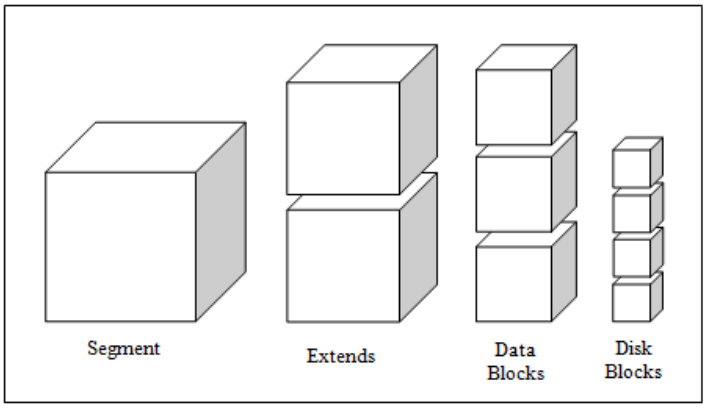


Gambar 3. Struktur Memori dan *Background* Proses *Oracle*

Sebuah *Oracle instance* dibentuk dari struktur memori yang dikenal dengan SGA dan *background process* yang menangani banyak pekerjaan dalam sebuah *instance* (Best). Beberapa *background process* yang umum adalah: (1) *System Monitor* (SMON). Bekerja saat *instance* diaktifkan setelah mengalami *crash* atau *failure*, tahap ini disebut dengan *crash recovery*. (2) *Process Monitor* (PMON). Melakukan proses pembersihan memori saat sebuah proses *user*

mengalami kegagalan. (3) *Database Writer* (DBW*n*). Menuliskan *block* yang telah dimodifikasi dari *database buffer cache* ke dalam *data file* pada *disk*. (4) *Checkpoint* (CKPT). Memperbarui semua *data file* dan *control file database* untuk menunjukan *checkpoint* yang paling baru. (5) *LogWriter* (LGWR). Menuliskan catatan *redo log* ke dalam *disk*. Saat *Oracle Data Guard* diaktifkan, LGWR juga dapat bertugas untuk mengirimkan *redo log file* ke *standby database* pada mode *maximum protection* atau *maximum availability*. (6) *Archiver* (ARC*n*). Salinan *file redo log* ke dalam penyimpanan *archival* saat terjadi *log switch*. *Archiver* juga dapat bertugas untuk mengirimkan *redo log file* ke *standby database* pada mode *maximum performance*.

Database object seperti tabel dan *index* disimpan sebagai *segment* dalam *tablespaces*. Setiap *segment* berisi satu atau lebih *extend*. Sebuah *extend* terdiri dari sekumpulan *data blocks* yang berdekatan, yang berarti setiap *extend* hanya akan berada dalam satu *data file*. *Data block* adalah unit terkecil dari I/O dalam basis data. Ukuran *data block* dapat ditentukan pada waktu pembuatan basis data. Ukuran *default*-nya adalah 8 KB. Jika basis data digunakan untuk aplikasi *data warehouse* yang mempunyai tabel dan *index* yang besar, maka ukuran *block* yang lebih besar akan sangat bermanfaat. *Redo log file* disimpan dalam ukuran *data block*, sehingga setiap kali pengiriman data dari *primary database* ke *standby database*, data yang dikirim bukan per *record*, melainkan sebuah *data block*.



Gambar 4. *Segment*, *Extends*, *Data Blocks*, dan *Disk Blocks*

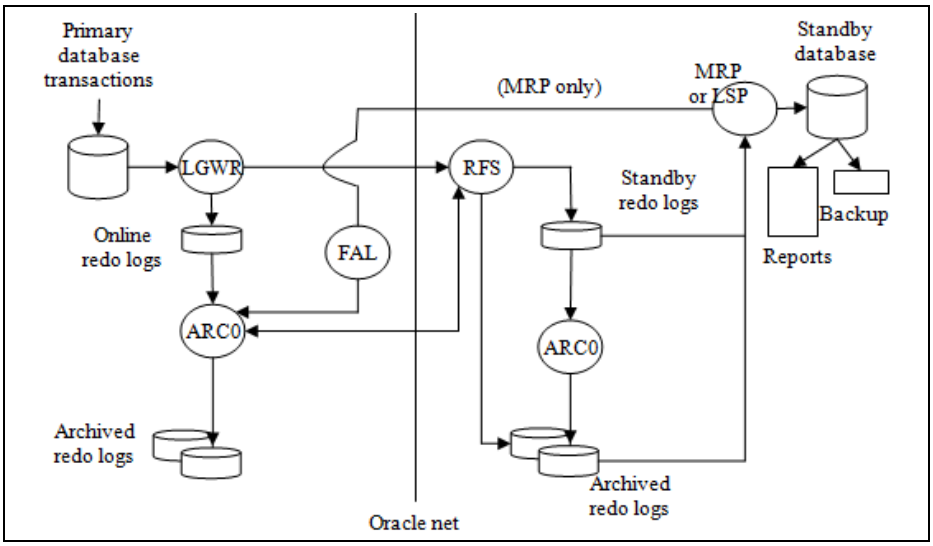
2.4. *Oracle Data Guard*

Oracle Data Guard adalah salah satu produk *Oracle* yang terdapat dalam instalasi *Oracle Database*. *Oracle Data Guard* adalah sebuah infrastruktur perangkat lunak yang digunakan untuk manajemen, monitoring, dan automatisasi yang berkerja bersama sebuah basis data produksi dan satu atau lebih *standby database* untuk melindungi data dari *failure*, *error*, dan *corruption* yang mungkin dapat mempengaruhi kinerja basis data.

Oracle Data Guard terbagi menjadi 2 bagian, yaitu *primary database* (pada bagian kiri gambar 5) dan *standby database* (pada bagian kanan gambar 5). *Primary database* bekerja secara aktif menerima semua proses transaksi dan mengirimkan data transaksi ke *standby database*, sedangkan *standby database* bekerja secara pasif dengan menerima data transaksi yang dikirimkan dari *primary database*. *Standby database* bekerja secara aktif (berubah menjadi *primary database*), saat *primary database* mengalami *switchover* atau *failover*, dan langsung bekerja tanpa membutuhkan waktu *recovery*. *Standby database* berfungsi untuk menerima *redo log* yang dikirimkan oleh *primary database*, *redo log* tersebut akan diterima, dan dieksekusi sendiri oleh *standby database* (Keesling). *Oracle* mempunyai dua tipe *standby database* yaitu *physical standby database* dan *logical standby database*.

Physical standby database secara fisik identik dengan *primary database*. *Database* struktur sama dengan *primary database* berdasarkan pada basis *block-for-block* (Keesling). *Physical standby database* di *update* berdasarkan *redo data* yang didapat dari *primary database*.

Database ini hanya dapat digunakan untuk *recovering data* dan *open* untuk laporan *read-only. Logical standby database* selalu bersinkronisasi dengan *primary database* dengan mengubah data dari *redo* yang dikirimkan oleh *primary database* menjadi *statement SQL* yang kemudian dijalankan dalam *standby database* (Keesling). Hal ini dapat dilakukan dengan menggunakan teknologi *Log Miner*. Tabel dalam *logical standby database* dapat digunakan secara simultan untuk *recovery* dan keperluan lain seperti laporan, *summation*, dan *query.*



Gambar 5. Arsitektur *Oracle Data Guard*

Pada *primary database*, *Data Guard log transport services* menggunakan beberapa proses berikut ini (Keesling): (1) Proses *log writer* (LGWR). LGWR menggumpulkan *redo information* transaksi dan meng-*update online redo logs*. Dalam mode *synchronous*, LGWR mengirimkan informasi *redo* langsung ke proses *remote file server* (RFS) pada *standby database* dan menunggu konfirmasi sebelum dijalankan. Dalam mode *asynchronous*, LGWR mengirimkan informasi *redo* langsung, tetapi tanpa menunggu konfirmasi. (2) Proses *archiver* (ARC*n*). ARC*n* adalah *copy* dari *redo logs* secara lokal untuk digunakan dalam *primary database recovery*. Proses ARC*n* juga dapat mengirim *redo stream* ke RFS disaat yang bersamaan dengan *archiving online log*. ARC*n* juga bertugas secara proaktif untuk mendeteksi dan menyelesaikan celah *log* pada semua *standby database*. (3) *Fetch archive log* (FAL) (hanya pada *physical standby database*). FAL menyediakan sebuah mekanisme *client-server* untuk menyelesaikan celah *log* yang dideteksi dengan *archived redo log* yang di-*generate* pada *primary database* dan didapat pada *standby database*. Proses ini berjalan hanya pada saat dibutuhkan dan *shutdown* langsung setelah tugasnya selesai.

Pada *standby database*, *Data Guard log apply services* menggunakan beberapa proses dibawah ini (Keesling): (1) Proses *Remote file server* (RFS). RFS menerima *redo information* dari *primary database*. RFS dapat menuliskan *redo* ke dalam *standby redo logs* atau *archived redo logs*. (2) Proses *archiver* (ARC*n*). Proses ARC*n* meng-*archive standby redo logs*. (3) *Managed recovery prosess* (MRP). MRP meng-*apply archieved redo log information* ke *physical standby database* (hanya untuk *physical standby database*), dengan menggunakan perintah *SQL* untuk *recovery "ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION"*. (4) *Logical standby process* (LSP). LSP mengontrol *apply* dari *archived redo log information* ke *logical standby database* (hanya untuk *logical standby database*).

Pada komputer *physical standby database* ada 2 mode yang bisa dipilih: (1) *Redo Apply*. Dalam mode ini, *log transport services* meng-*archive log* ke *standby database*, dan *log apply services* secara otomatis meng-*apply log* ini. Database dalam *state MOUNT*, data tidak dapat diakses. (2) *Open read-only*. Digunakan untuk tujuan *reporting*. *Log apply services* tidak

dapat meng-*apply archive redo log* ke *standby database* pada mode ini, tetapi *primary database* tetap mengirimkan *redo* ke *standby database*. Untuk komputer *logical standby database*, mempunyai kebebasan untuk memilih mode *Open read/write*. Pada mode ini, *log apply services* tetap memanajemen informasi *log* dari *archived redo log*. Mode *open* bisa digunakan untuk *reporting*.

Oracle Data Guard menyediakan mode *maximum protection*, *maximum availability*, dan *maximum performance* untuk membantu DBA dalam menjaga keamanan data. Dalam beberapa situasi, data sebuah bisnis menjadi sangat penting dan tidak boleh hilang. Sedangkan beberapa aplikasi yang lain membutuhkan *maximum database performance* dan dapat mentolerir kemungkinan kehilangan data.

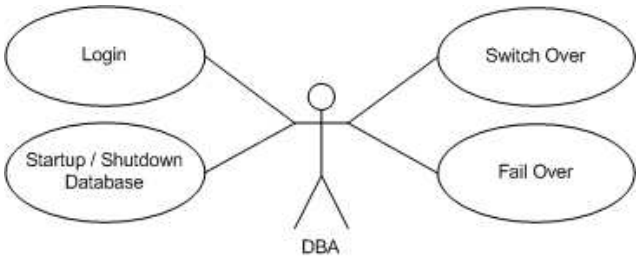
Mode *Maximum Protection* menjamin tidak ada data yang hilang kecuali *primary database fail*. Untuk mengkonfigurasi tingkat proteksi ini, dibutuhkan *redo data* untuk me-*recover* setiap transaksi yang harus dituliskan ke dalam *local online redo log* dan *standby redo log* pada sedikitnya satu *standby database* sebelum transaksi di *commit*. Untuk menjamin data tidak pernah hilang, *primary database shut down* jika ada kesalahan yang terjadi pada penulisan *redo stream* ke sedikitnya satu *remote standby redo log*. Mode *Maximum Availability* menyediakan tingkat proteksi keamanan data yang paling tinggi tanpa mengetahui keadaan *primary database*. Sama seperti *maximum protection mode*, sebuah transaksi tidak akan pernah *commit* sampai *redo* yang dibutuhkan untuk me-*recover* transaksi tersebut dituliskan ke dalam *redo stream* dan paling sedikit pada satu *remote standby redo log*. *Primary database* tidak akan *shut down* jika terjadi kegagalan dalam penulisan *redo stream* ke dalam *standby redo log*. Pengiriman *redo log file* akan berhenti saat terjadi kegagalan penulisan. *Gap* antara *redo log* akan diselesaikan pada saat pebaikan kegagalan tersebut. Mode ini menjamin tidak ada data yang hilang jika *primary database* mengalami kegagalan, hanya pengiriman *redo data* akan berhenti sementara sampai adanya perbaikan. Mode *Maximum Performance* adalah *default* dari *Data Guard* yang menyediakan proteksi tertinggi yang dapat dilakukan tanpa mengurangi performa dari *primary database*. Mode ini mengijinkan sebuah transaksi untuk *commit* saat *redo data* yang dibutuhkan untuk *recover* transaksi tersebut ditulis pada minimal satu *standby database*, tetapi *redo stream* ditulis secara *asynchronous* dengan sebuah kesepakatan transaksi tersebut akan membuat *redo data*. Mode ini digunakan pada sistem dengan *bandwidth* yang cukup, dengan mengurangi *traffic* agar performa *database* tetap terjaga.

Operasi *switchover* dan *failover* tidak akan berjalan secara otomatis. DBA yang harus menginisiasikan operasi *switchover* dan *failover* dengan menggunakan *SQL statement* atau dengan menggunakan GUI *Data Guard* atau dengan *command-line interface* (CLI) *Data Guard broker*.

3. Metodologi Penelitian

3.1. Diagram *Use Case* Untuk DBA

Selain aktor-aktor dalam Simulasi SIA-SAT, terdapat aktor tambahan yaitu DBA (*database administrator*) yang mengurusi basis data. Gambar 6 adalah diagram *use case* untuk aktor DBA, dengan beberapa *use case* yang penting dalam penelitian ini.

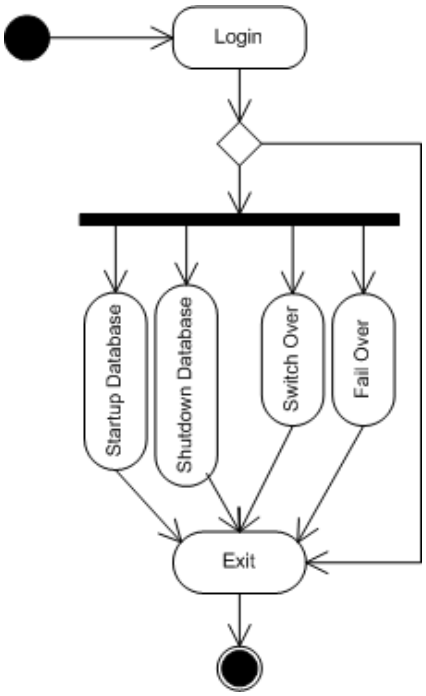


Gambar 6. Diagram *Use Case* DBA

3.2. Diagram *Activity* DBA

DBA adalah seorang *user* yang memiliki kemampuan administrasi basis data. Secara umum mereka bertugas untuk membuat, dan merawat basis data. Aktifitas mereka dalam basis data sangat banyak, dari mulai melakukan instalasi, membuat tabel, membuat *user*, mengawasi kondisi basis data, melakukan *backup*, dan melakukan *restore* basis data.

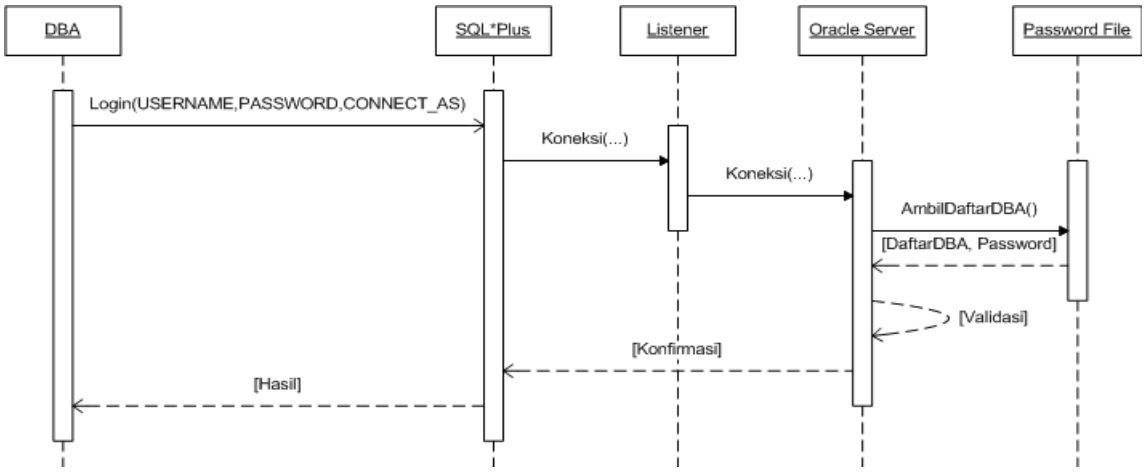
Gambar 7 hanya menggambarkan sebagian kecil aktifitas yang terjadi dalam *user* DBA, yaitu: *startup database*, *shutdown database*, *switchover*, dan *failover*.



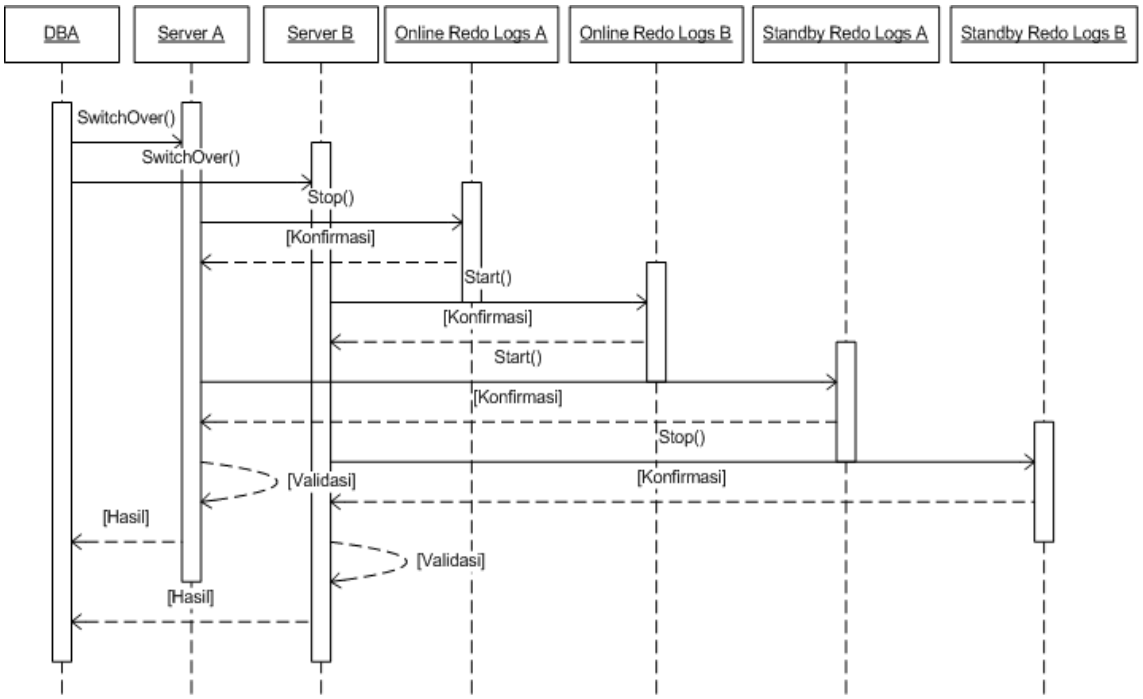
Gambar 7. Diagram *Activity* DBA

3.3. Diagram *Sequence* DBA

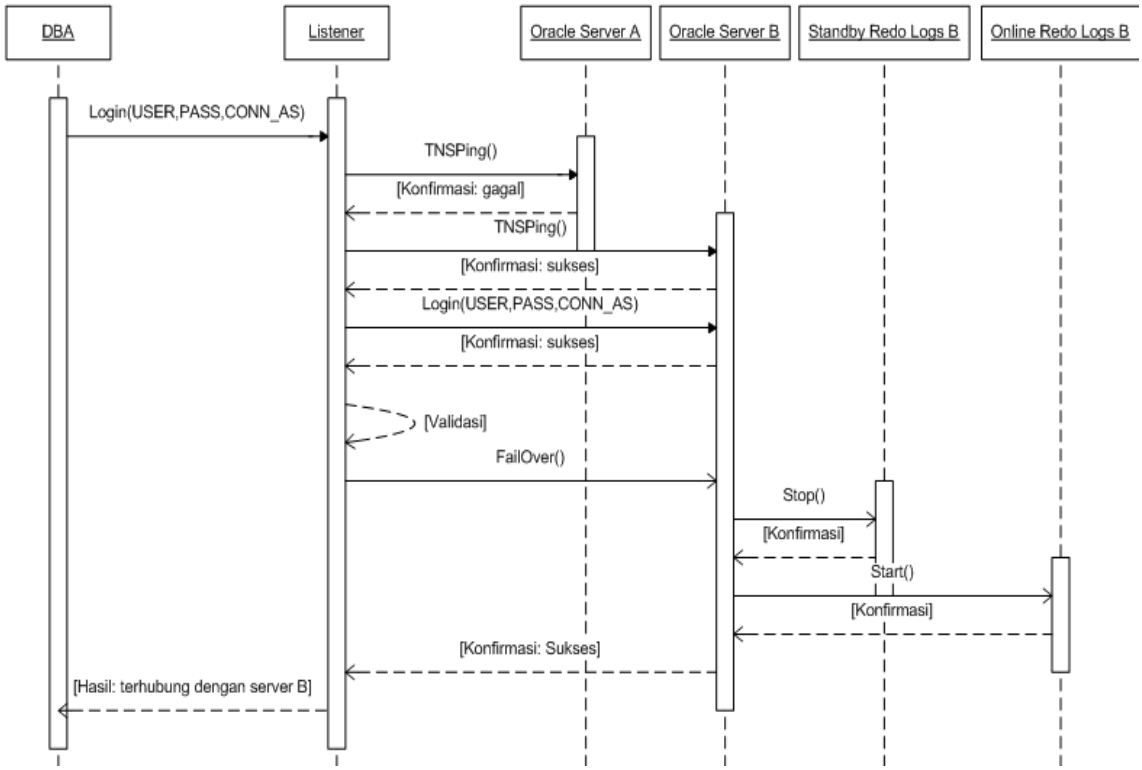
Pada diagram *sequence* DBA, terdapat 4 diagram, yaitu: diagram *sequence login*, diagram *sequence startup* dan *shutdown immediate*, diagram *sequence switchover*, dan diagram *sequence failover*. Diagram tersebut mengambarkan urut-urutan proses secara umum.



Gambar 8. Diagram *Sequence Login*



Gambar 9. Diagram *Sequence Switchover*

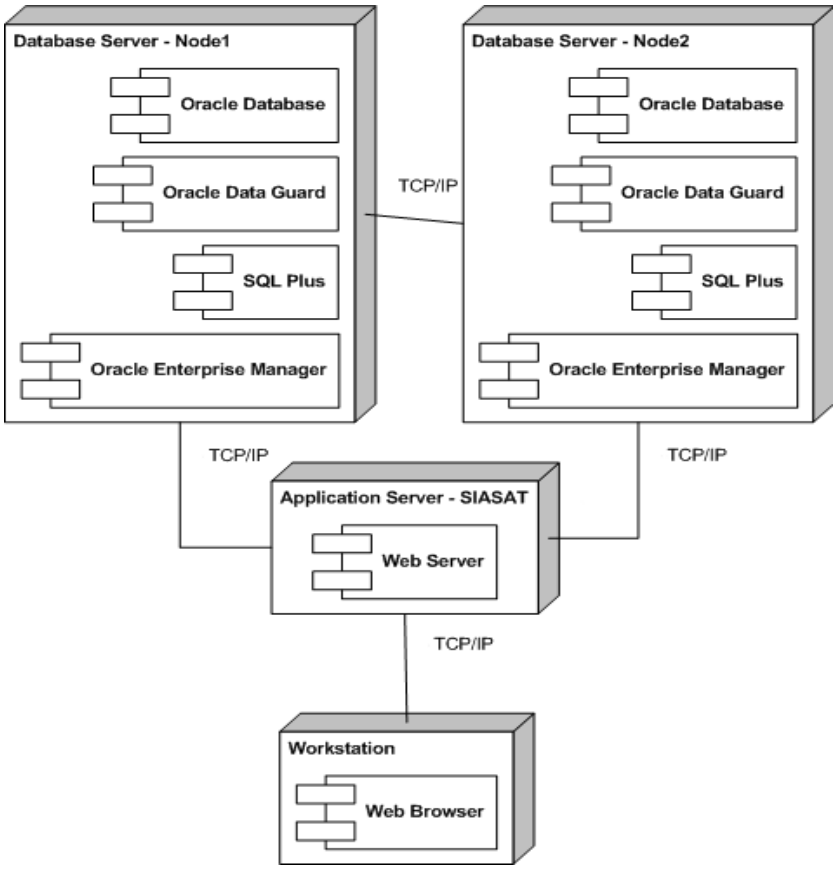


Gambar 10. Diagram *Sequence Failover*

3.4. Diagram *Deployment*

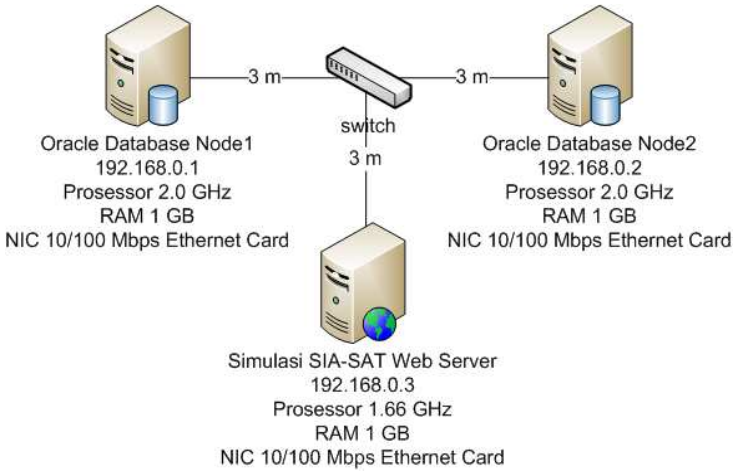
Tujuan dari *Oracle Data Guard* adalah untuk mencapai sebuah sistem yang *high availability*. Untuk mengimplementasikan sistem ini, diperlukan beberapa persiapan yang harus

dilakukan. Secara teknis, pada tahap ini yang harus dipersiapkan adalah *hardware* dan *software* untuk membangun sistem ini, yaitu server, perangkat jaringan, sistem operasi, dan *software*.



Gambar 11. Diagram *Deployment*

Rancangan sistem merupakan dasar bagi sistem ini. *Logical design* dapat diartikan sebagai konsep rancangan yang biasanya meliputi struktur jaringan, jumlah server yang akan digunakan, dan lain-lain. *Physical design* adalah rancangan yang lebih spesifik, contohnya adalah merk server atau NIC. Gambar 12 adalah rancangan topologi sistem *Oracle Data Guard* dan SIA-SAT.



Gambar 12. Topologi Rancangan Sistem *Oracle Data Guard* Dengan *Web Server* SIA-SAT

4. Pengujian dan Pembahasan

Untuk menguji apakah node1 telah terhubung dengan node2, dengan melihat status *log* sekarang, dan nomor *log* pada node1.

```
SQL> select * from v$log;

    GROUP#    THREAD#  SEQUENCE#      BYTES    MEMBERS ARC STATUS
---------- ---------- ---------- ---------- ---------- --- ---------
FIRST_CHANGE# FIRST_TIM
------------- ---------
         1          1          5   52428800          1 NO  INACTIVE
       532172 14-JAN-09

         2          1          6   52428800          1 NO  CURRENT
       557887 14-JAN-09

         3          1          4   52428800          1 NO  INACTIVE
       508917 14-JAN-09
```

Gambar 13. Status Log File Pada Node1

Status *log sequence* berada pada no 6 karena statusnya pada saat ini adalah "*CURRENT*". *Log* ini disimpan dalam *control file*, dalam *database* ini mempunyai 3 buah *control file*. *Control file* selalu berganti (*switch*) jika terjadi dua kondisi, yakni jika *control file* sudah penuh, (2) Atau terjadi *log switch.* Jika kita melakukan perintah *ALTER SYSTEM SWITCH LOGFILE*, maka *log sequence* akan bertambah menjadi 7, hal ini dapat dilihat pada gambar 14.

```
SQL> select * from v$log;

    GROUP#    THREAD#  SEQUENCE#      BYTES    MEMBERS ARC STATUS
---------- ---------- ---------- ---------- ---------- --- ----------
FIRST_CHANGE# FIRST_TIM
------------- ---------
         1          1          5   52428800          1 NO  INACTIVE
       532172 14-JAN-09

         2          1          6   52428800          1 NO  ACTIVE
       557887 14-JAN-09

         3          1          7   52428800          1 NO  CURRENT
       558240 14-JAN-09
```

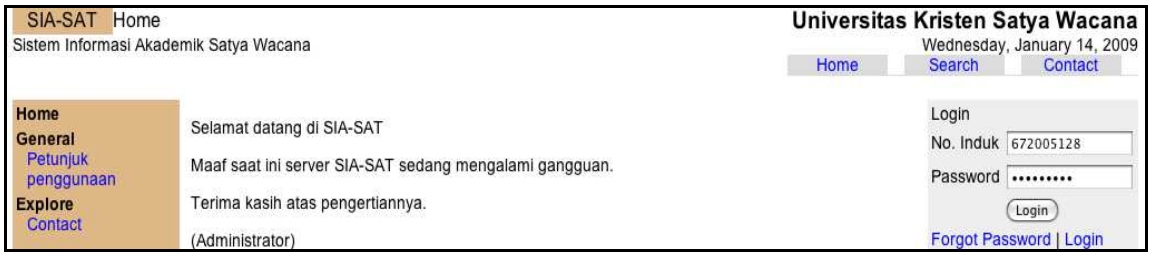
Gambar 14. Status *Log File* Pada Node1 dan Node2

Pada *standby database* (node2), basis data tidak dapat dipergunakan karena harus menunggu data dari node1, namun basis data dapat dibuka dengan status *read only*. Berikut adalah isi dari tabel admin pada node2.

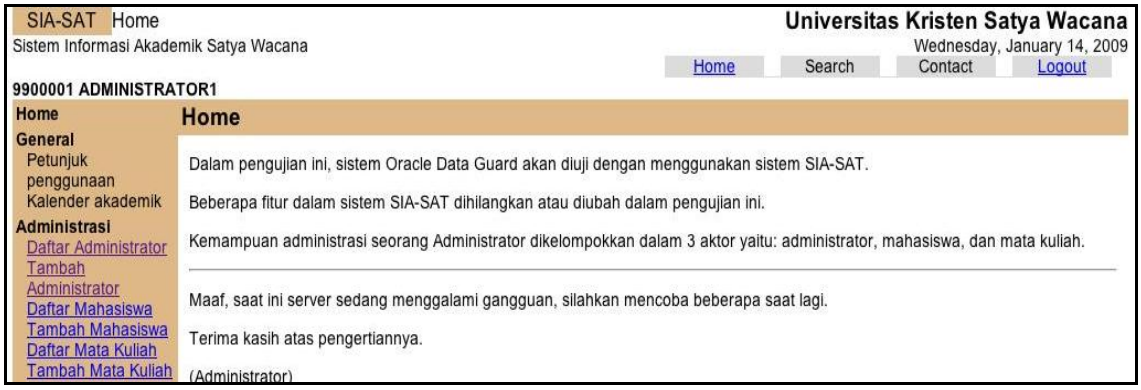
```
SQL> conn user/user
'Connected.
SQL>select * from admin;

NIK     PASSWORD
------- --------------------------------------------
NAMA
--------------------------------------------
9900001 admin
administrator1

9900002 admin
administrator2
```

Gambar 15. Isi Tabel Admin Pada Node2

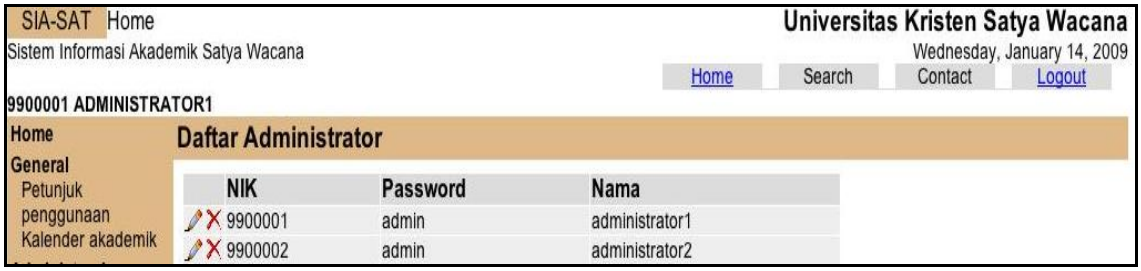
Untuk pengujian *failover*, maka dalam pengujian ini, penulis melakukan pemutusan *network*, sehingga node1 terpisah dari jaringan. Pada SIA-SAT akan muncul pada halaman login.



Gambar 16. Halaman Login Pada Saat *Database* Beralih



Gambar 17. Halaman Admin Saat *Database* Beralih



Gambar 18. Halaman Daftar Administrator Setelah Terjadi *Failover*

Failover membutuhkan beberapa saat untuk dapat membuat node2 aktif untuk menggantikan node1. Simulasi SIA-SAT akan sementara berhenti, dikarenakan tidak dapat mengakses basis data hingga node2 aktif. Setelah node2 aktif, *web server* akan dapat mengakses basis data kembali, dan tampilan kembali seperti pada gambar 18.

4.1. Percobaan *Failover*

Percobaan ini dilakukan dalam beberapa proses dalam web, kemudian *failover* dilakukan. Parameter yang menjadi pembanding adalah waktu, kondisi data, kondisi *web server*. *Failover* akan dilakukan dengan memutuskan jaringan *primary database*. Pemutusan jaringan dipilih karena jaringan merupakan salah satu faktor yang penting dalam sistem ini yang menjadi penghubung antara kedua basis data, dan pemutusan jaringan merupakan percobaan yang aman bagi kondisi komputer, sehingga kondisi fisik komputer tidak terganggu.

Percobaan ini dilakukan saat seorang *user* SIA-SAT melakukan proses *insert* ke *database*, kemudian langsung dilakukan *failover* beberapa detik saat hasil *insert* telah muncul. Percobaan ini dilakukan sebanyak 10 kali, sehingga didapat waktu rata-rata 1 menit 26 detik.

Tabel 2. Hasil Percobaan *Failover*

Menit	Detik	Availability
1	30	99.999707%
1	28	99.999714%
1	26	99.999720%
1	27	99.99717%
1	27	99.99717%
1	25	99.99724%
1	24	99.99727%
1	28	99.999714%
1	21	99.99737%
1	28	99.999714%
Rata-rata	1 menit 26 detik	99.999720%

4.2. Percobaan *Failover* Tanpa *Data Guard*

Maksud dari *failover* tanpa *data guard* adalah dengan me-*recovery* data pada node2 secara manual dari *backup* terakhir pada node1, lalu pengalihan koneksi web server ke node2. Pada saat kondisi data awal, yaitu pada tabel admin hanya memiliki 1 data administrator, node1 akan di-*backup*, dan *backup file* akan dipindah ke node2. *Backup* dilakukan dengan *tool* RMAN (*Recovery Manager*).

Node1 akan ditambah 1 data administrator2, dan akan matikan, sehingga web server harus mengalihkan koneksi ke node2. Pada node2, dilakukan *recovery* dengan menggunakan *tool* RMAN. Waktu rata-rata yang diperlukan untuk *recover database* ini adalah 10 menit 57 detik.

Tabel 3. Hasil Percobaan *Failover* Tanpa *Oracle Data Guard*

Menit	Detik	Availability
10	43	99.997910%
11	6	99.997835%
10	9	99.998020%
11	12	99.997815%
10	32	99.997945%
11	13	99.997812%
10	35	99.997936%
11	42	99.997718%
10	53	99.997877%
11	28	99.997763%
Rata-rata	10 menit 57 detik	99.997864%

5. Kesimpulan

Dengan menggunakan *Oracle Data Guard*, proses perpindahan basis data saat terjadi *failure*, tidak membutuhkan proses *recovery*, sehingga tidak menambah waktu *downtime*. Hal ini dapat terjadi karena data langsung ditambahkan ke dalam *data file*. Terjadi peningkatan waktu *uptime* jika menggunakan *Oracle Data Guard*. Sistem ini menghemat waktu untuk *recovery* data. Dengan peningkatan waktu *uptime*, waktu pelayanan *database* pun akan bertambah. Data akan lebih aman jika menggunakan *physical standby database*, karena berfungsi sebagai replika dari *primary database*. Dibandingkan dengan *backup* yang biasa, data tidak selalu ter-*update* setiap saat.

Referensi

Best, T., Billings, M.J. 2005. *Oracle Database 10g: Administration Workshop I.* California.

Bharwara, D.I. 2001. *Cluster High Availability Pada Server LTSP*. Salatiga: Fakultas Teknik Elektro Universitas Kristen Satya Wacana.

Fathansyah. 2004. *Sistem Basis Data*. Bandung: Informatika.

Greenberg, N. 2004. *Oracle Database 10g: SQL Fundamental I*. California.

Han, Y. 2005. *An Integrated High Availability Computing Platform*. Arizona: Academic Research Library University of Arizona.

Keesling, D., Van Dyke, R. 2005. *Oracle Database 10g: Data Guard Administration*. California.

Koopmann, J.F. 2007. *Achieving High Availability on a Linux Two-Node Cluster Using Oracle's Maximum Availability Architecture*. Canada: LSI Logic Corporation.

Mannino, M.V. 2004. *Database design, application development, and administration*. New York: Mcgraw-Hill.

Saputra, H., Christian. 2008. *Perancangan dan Implementasi Basis Data Terdistribusi Menggunakan Arsitektur SAN (Storage Area Network) pada Rekam Medis on-line*. Salatiga: Fakultas Teknologi Informasi Universitas Kristen Satya Wacana.